Control Theory Tech, Vol. 13, No. 1, pp. 55–69, February 2015

Control Theory and Technology

http://link.springer.com/journal/11768

Discrete-time dynamic graphical games: model-free reinforcement learning solution

Mohammed I. ABOUHEAF [1†], Frank L. LEWIS [2,3], Magdi S. MAHMOUD [1], Dariusz G. MIKULSKI [4]

1. *Systems Engineering Department, King Fahd University of Petroleum & Mineral, Dhahran - 31261, Saudi Arabia;*

2. *UTA Research Institute, University of Texas at Arlington, Fort Worth, Texas, U.S.A.;*

3. *State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang Liaoning 110819, China;*

4. *Ground Vehicle Robotics (GVR), U.S. Army TARDEC, Warren MI, U.S.A.*

Received 31 December 2013; revised 2 January 2015; accepted 15 January 2015

Abstract

This paper introduces a model-free reinforcement learning technique that is used to solve a class of dynamic games known as dynamic graphical games. The graphical game results from multi-agent dynamical systems, where pinning control is used to make all the agents synchronize to the state of a command generator or a leader agent. Novel coupled Bellman equations and Hamiltonian functions are developed for the dynamic graphical games. The Hamiltonian mechanics are used to derive the necessary conditions for optimality. The solution for the dynamic graphical game is given in terms of the solution to a set of coupled Hamilton-Jacobi-Bellman equations developed herein. Nash equilibrium solution for the graphical game is given in terms of the solution to the underlying coupled Hamilton-Jacobi-Bellman equations. An online model-free policy iteration algorithm is developed to learn the Nash solution for the dynamic graphical game. This algorithm does not require any knowledge of the agents' dynamics. A proof of convergence for this multi-agent learning algorithm is given under mild assumption about the inter-connectivity properties of the graph. A gradient descent technique with critic network structures is used to implement the policy iteration algorithm to solve the graphical game online in real-time.

Keywords: Dynamic graphical games, Nash equilibrium, discrete mechanics, optimal control, model-free reinforcement learning, policy iteration

DOI 10.1007/s11768-015-3203-x

†Corresponding author.

E-mail: abouheaf@kfupm.edu.sa. Tel.: +966 (13) 860 2968; fax: +966 (13) 860 2965. King Fahd University of Petroleum & Mineral, P.O. Box 1956, Dhahran - 31261, Saudi Arabia.

This work was supported by the Deanship of Scientific Research at King Fahd University of Petroleum & Minerals Project (No. JF141002), the National Science Foundation (No. ECCS-1405173), the Office of Naval Research (Nos. N000141310562, N000141410718), the U.S. Army Research Office (No. W911NF-11-D-0001), the National Natural Science Foundation of China (No. 61120106011), and the Project 111 from the Ministry of Education of China (No. B08015).

1 Introduction

This paper develops an online model-free policy iteration solution [1, 2] for a class of discrete-time dynamic graphical games developed in [3]. The information flow between the agents is governed by a communication graph. Continuous-time differential graphical games have been developed in [4]. Novel model-free policy iteration algorithm is developed to learn Nash solution for graphical game in real-time. This paper brings together cooperative control, optimal control, game theory, and reinforcement learning techniques to find online solutions for the graphical game.

The optimal control theory uses the Hamilton-Jacobi-Bellman (HJB) equation whose solution is the optimal cost-to-go function [5, 6]. Discrete-time canonical forms for the Hamiltonian functions are found in [7–9]. The cooperative control problems involve the consensus control problems and the synchronization control problems [10–16]. The agents synchronize to uncontrollable node dynamics in the cooperative consensus problem. While in the synchronization control problem, the control protocols are designed such that each agent reaches the same state [17–21].

Game theory provides a solution framework for multi-agent control problems [22]. Non-cooperative dynamic game theory provides an environment for formulating multi-player decision control problems [23]. Each agent finds its optimal control policy through optimizing its performance index independently [23]. Offline solutions for the games are given in terms of the respective coupled Hamilton-Jacobi (HJ) equations [23–25].

Approximate dynamic programming (ADP) is an approach to solve the dynamical programming problems [1, 2, 26–31]. ADP combines adaptive critics and reinforcement learning (RL) techniques, with dynamic programming [2]. ADP techniques are developed to solve the optimal control problem online in [2] and offline in [29]. Morimoto et al used the concept of Q-learning to solve the differential dynamic programming [32]. Landelius, used the action dependent heuristic dynamic programming (ADHDP) to solve the linear quadratic optimal control problem and showed that the solution is equivalent to iterating the underlying Riccati equations [33]. RL is concerned with learning from interaction in a dynamic environment [34, 35]. RL algorithms are used to learn the optimal control solutions for dynamic systems in real-time [1, 2, 34, 36, 37]. These algorithms involve policy iteration (PI) or value iteration (VI) techniques [29, 36, 38–42]. New policy iteration approach employed ADP to find online solutions for the continuous-time Riccati equations in [43]. Policy iteration solution for the adaptive optimal control problem can be obtained by relaxing the HJB equation to the equivalent optimization problem [44]. RL algorithms are used to solve multi-player games for finite-state systems in [40–42] and to learn online in real-time the solutions for the optimal control problems of the differential games in [36–38, 45, 46]. Actor-critic neural network structures are used to solve the graphical game using heuristic dynamic programming (HDP) in real-time [3].

In this paper, the dynamic graphical game developed herein, is a special case of standard dynamic game [23] and explicitly captures the structure of the communication graph topology. The ADHDP structure for single agent [2] is extended for the case of the dynamic graphical games and it is used to solve the game in a distributed fashion unlike [23]. Usually, offline methods are employed to find Nash solutions for the games in terms of the coupled HJ equations (which are difficult to solve) [23–25]. Herein an online adaptive learning solution for the graphical game is given in terms of the solution to a set of novel coupled graphical game Hamiltonian functions and Bellman equations. Policy iteration convergence proof for the graphical game is given under mild condition about the graph interconnectivity. In [42], the Q learning update rule will converge to the optimal response Q-function as long as all the other agents converge in behavior. The developed online adaptive learning solution allows model-free tuning of the critic networks, while partial knowledge about the game was required in [3], [4], and [37].

The paper is organized as follows. Section 2 reviews the synchronization control problem for multi-agent systems on graphs. Section 3 formulates the dynamic graphical game in terms of the coupled Bellman equations and the respective Hamiltonian functions. This section finds the solution for the graphical game in terms of the solution to a set of coupled HJB equations. Section 4 shows that, the Nash solution for the graphical game is given in terms of the underlying coupled (HJB) equations. Section 5 develops an online adaptive model-free policy iteration algorithm to solve the dynamic graphical game in real-time along with its convergence proof. Section 6 implements the online policy iteration algorithm using critic network structures.

2 Synchronization of multi-agent systems on graphs

This section reviews the synchronization control problem on communication graphs.

2.1 Graphs

The directed graph $\hat{G}$ is defined as the pair $\hat{G} = (V, E)$ with a nonempty finite set of N vertices or agents $V = \{v_1, \ldots, v_N\}$ and a set of edges $E \subseteq V \times V$ [15]. The connectivity matrix E is defined such that $E = [e_{ij}]$ with $e_{ij} > 0$ if $(v_j, v_i) \in E$ and $e_{ij} = 0$ otherwise. The set of the neighbors of every agent v_i is $N_i = \{v_j : (v_j, v_i) \in E\}$. Define the diagonal in-degree matrix as $D = \text{diag}\{d_i\}$, with $d_i = \sum_{j \in N_i} e_{ij}$ the weighted in-degree of agent i. The graph Laplacian matrix L is defined as $L = D - E$ [15].

2.2 Synchronization and tracking error dynamics

The dynamics of each agent i is given by

$$x_i(k+1) = Ax_i(k) + B_i u_i(k), \tag{1}$$

where $x_i(k) \in \mathbb{R}^n$ is the state vector of agent i, and $u_i(k) \in \mathbb{R}^{m_i}$ is the control input vector for agent i.

A leader agent v_0 has the dynamics [47] $x_0(k) \in \mathbb{R}^n$ given by

$$x_0(k+1) = Ax_0(k). \tag{2}$$

The objective of the synchronization problem is to design the control inputs $u_i(k)$, using local neighbor information, so that all agents synchronize to the leader's dynamics, that is $\lim_{k\to\infty} \|x_i(k) - x_0(k)\| = 0, \forall i$. The leader is pinned to a small percentage of the agents in the graph.

To study synchronization on graphs, define the local neighborhood tracking error [48] $\varepsilon_i(k) \in \mathbb{R}^n$ for each agent i as

$$\varepsilon_i(k) = \sum_{j \in N_i} e_{ij}(x_j(k) - x_i(k)) + g_i(x_0(k) - x_i(k)), \tag{3}$$

where g_i is the pinning gain of agent i, which is nonzero $g_i > 0$ if agent i is coupled to the leader agent x_0 [18].

The overall tracking error vector $\varepsilon = [\varepsilon_1^{\mathrm{T}} \ \varepsilon_2^{\mathrm{T}} \ \cdots \ \varepsilon_N^{\mathrm{T}}]^{\mathrm{T}}$ is given by

$$\varepsilon(k) = -((L + G) \otimes I_n)\eta(k), \tag{4}$$

where $G = \text{diag}\{g_i\} \in \mathbb{R}^{N \times N}$ is a diagonal matrix of the pinning gains. The global synchronization error vector η [13] is given by

$$\eta(k) = (x(k) - \underline{x}_0(k)) \ \in \mathbb{R}^{nN}, \tag{5}$$

where $x \in \mathbb{R}^{nN}$ is the global state vector with $\underline{x}_0 = \underline{I} x_0 \in \mathbb{R}^{nN}$, $\underline{I} = \underline{1} \otimes I_n \in \mathbb{R}^{nN \times n}$ and $\underline{1}$ is the N-vector of ones.

The graph is assumed to be strongly connected and the pinning gain is $g_i > 0$ for at least one agent i, then the graph matrix $(L + G)$ is nonsingular [48]. The synchronization error is bounded such that

$$\|\eta(k)\| \leqslant \frac{\|\varepsilon(k)\|}{\underline{\sigma}(L + G)}, \tag{6}$$

where $\underline{\sigma}(\cdot)$ and $\bar{\sigma}(\cdot)$ denote the minimum and the maximum singular values of a matrix, respectively. For simplifying the notation, $x_i(k)$ is written as x_{ik}, and so on, when the time index k is clear.

Our objective is to minimize the local neighborhood tracking errors $\varepsilon_i(k)$, which in view of (6) will guarantee approximate synchronization.

The local neighborhood tracking error dynamics for agent i are given by

$$\begin{aligned} \varepsilon_{i(k+1)} &\equiv f_i(\varepsilon_{ik}, u_{ik}, u_{-ik}) \\ &= A\varepsilon_{ik} - (d_i + g_i)B_i u_{ik} + \sum_{j \in N_i} e_{ij} B_j u_{jk}, \end{aligned} \tag{7}$$

where u_{-i} are the control actions of the neighbors of each agent i $u_{-i} = \{u_j | j \in N_i\}$.

Define the group of control actions of the neighbors of each agent i and the control actions of the neighbors to the neighbors of each agent i as $u_{-i,-\{-i\}} = \{u_j | j \in N_i, N_{\{-i\}}\}$ and the actions of all the agents in the graph excluding i as $u_{\bar{i}} = \{u_j | j \in N, j \neq i\}$.

3 Multi-player cooperative games on graphs

In this section, solutions for the dynamic games on graphs are developed. These dynamic interacting games are based on the error systems (7), which are locally coupled in the sense that they are driven by the agent's control actions and those of its neighbors. The solution for the dynamic graphical game is given in terms of the solution to a set of novel coupled Hamiltonian functions and Bellman equations developed herein. The ADHDP structure for single agent [2] is extended to solve the dynamic graphical game without knowing any of the agents' dynamics.

3.1 Graphical games: performance evaluation

Graphical games are based on the interactions of each agent i with the other players in graph. The local neighborhood dynamics (7) arise from the nature of synchronization problem for dynamic systems on communication graphs. Therefore, in order to define a dynamic graphical game, the local performance index for each agent i is written as

$$J_i = \sum_{k=0}^{\infty} U_i(\varepsilon_{ik}, u_{ik}, u_{-ik}) \tag{8}$$

and the utility function U_i for each agent i is given by

$$\begin{aligned} &U_i(\varepsilon_{ik}, u_{ik}, u_{-ik}) \\ &= \frac{1}{2}(\varepsilon_{ik}^{\mathrm{T}} Q_{ii} \varepsilon_{ik} + u_{ik}^{\mathrm{T}} R_{ii} u_{ik} + \sum_{j \in N_i} u_{jk}^{\mathrm{T}} R_{ij} u_{jk}), \end{aligned} \tag{9}$$

where $Q_{ii} \geqslant 0 \in \mathbb{R}^{n \times n}$, $R_{ii} > 0 \in \mathbb{R}^{m_i \times m_i}$, and $R_{ij} > 0 \in \mathbb{R}^{m_j \times m_j}$ are symmetric time-invariant weighting matrices.

For the multi-player graphical game, it is desired to determine the optimal non-cooperative solutions such that the following distributed coupled optimizations are solved simultaneously,

$$J_i^* = \min_{u_{ik}} \sum_{k=0}^{\infty} U_i(\varepsilon_{ik}, u_{ik}, u_{-ik}^*), \quad \forall i \in N, \tag{10}$$

where u_{-ik}^* denotes the optimal policies in the neighboring policies. In the subsequent sections we will use information of the neighboring states as well to define a prototypical game theoretic framework that solves our desired problem and the optimal policies are proved to form a Nash equilibrium. Simulation results will show that this formulation is more effective than just using the current agent's state.

Given fixed policies (μ_{il}, μ_{-il}) of agent i and its neighbors, the value function for each agent i is given by

$$V_i(\bar{\varepsilon}_{ik}) = \sum_{l=k}^{\infty} U_i(\varepsilon_{il}, \mu_{il}, \mu_{-il}), \tag{11}$$

where $\bar{\varepsilon}_{ik}$ is a vector of the state ε_{ik} of agent i and the states of its neighbors ε_{-ik}.

Remark 1 The performance index (8) measures the performance of each agent i. The value function for each agent i (11) captures local information. Thus, the solution structure of the value function will be given in terms of the local vector $\bar{\varepsilon}_{ik}$. This value structure will be used in the mathematical setup for the graphical game.

Definition 1 The control policies $\bar{u}_i = \{u_{ik}\}_{k=0}^{\infty}, \forall i \in N$ are said to be admissible if they stabilize (7) and guarantee that $V_i(\bar{\varepsilon}_{ik}), \forall i$ are finite.

Definition 2 The dynamic graphical game with local dynamics (7) and performance indices (8) is well-formed if $R_{ij} \neq 0 \rightleftharpoons e_{ij} \in E$.

3.2 Hamiltonian function for graphical games

Given the dynamics (7) and the performance indices (8), define the Hamiltonian function [6] of each agent i as

$$\begin{aligned} &H_i(\bar{\varepsilon}_{ik}, \lambda_{i(k+1)}, u_{ik}, u_{-ik}) \\ &= \lambda_{i(k+1)}^{\mathrm{T}}(\bar{\varepsilon}_{i(k+1)})\bar{\varepsilon}_{i(k+1)} + U_i(\varepsilon_{ik}, u_{ik}, u_{-ik}), \end{aligned} \tag{12}$$

where $\lambda_{ik} \equiv \lambda_i(k)$ is the costate variable of each agent i, and $\bar{\varepsilon}_{ik} = \tilde{Z}_{ik}\varepsilon_k = \begin{bmatrix} 0 \cdots [I_n]_{ii} \cdots 0 \\ 0 \cdots [I_n]_{ij} \cdots 0 \end{bmatrix} \varepsilon_k \in \mathbb{R}^{nN_{i,j}}$. $N_{i,j}$ denotes the number of each agent i and its neighbors j.

Denote the stationary admissible policy for each agent i by $u_{ik} = \pi_{ik} = \pi_i(\bar{\varepsilon}_{ik})$. Thus the Hamiltonian given fixed stationary policies for the neighbors is written such that

$$\begin{aligned} &H_i^{\pi}(\bar{\varepsilon}_{ik}, \lambda_{i(k+1)}, u_{ik}, \pi_{-ik}) \\ &= \lambda_{i(k+1)}^{\mathrm{T}}(\bar{\varepsilon}_{i(k+1)}) + U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}). \end{aligned} \tag{13}$$

The optimal control policy based on the Hamiltonian (13) is given by applying the stationarity condition [6] $\dfrac{\partial H_i}{\partial u_{ik}} = 0$ such that

$$u_{ik}^* = \arg\min_{u_{ik}}(H_i^{\pi}(\bar{\varepsilon}_{ik}, \lambda_{i(k+1)}, u_{ik}, \pi_{-ik})). \tag{14}$$

Then,

$$u_{ik}^* = M_i \lambda_{i(k+1)}, \tag{15}$$

where $M_i = R_{ii}^{-1}([\cdots (g_i + d_i) \cdots - e_{ji} \cdots] \otimes B_i^{\mathrm{T}})$, $e_{ji}, j \in N_i$ are the out neighbors connectivity weights.

Applying the optimality condition $\dfrac{\partial H_i}{\partial \bar{\varepsilon}_{ik}} = \lambda_{ik}$ [6] yields the costate equation such that

$$\lambda_{ik} = \bar{A}_i^{\mathrm{T}} \lambda_{i(k+1)} + \bar{Q}_i \bar{\varepsilon}_{ik}, \tag{16}$$

where $\bar{A}_i = \mathrm{diag}\{A, \ldots, A\}$, $\bar{Q}_i = \mathrm{diag}\{\ldots, Q_{ii}, \ldots\} \in \mathbb{R}^{nN_{i,j} \times nN_{i,j}}$.

3.3 Bellman equation for graphical games

Herein Bellman optimality equations are developed to solve the graphical games. The value function (11) given stationary admissible policies yields the dynamic

graphical game Bellman equations such that

$$V_i^{\pi}(\bar{\varepsilon}_{ik}) = \frac{1}{2}(\varepsilon_{ik}^{\mathrm{T}} Q_{ii} \varepsilon_{ik} + \pi_{ik}^{\mathrm{T}} R_{ii} \pi_{ik} + \sum_{j \in N_i} \pi_{jk}^{\mathrm{T}} R_{ij} \pi_{jk}) + V_i^{\pi}(\bar{\varepsilon}_{i(k+1)}) \tag{17}$$

with initial conditions $V_i^{\pi}(\mathbf{0}) = 0$.

Define the difference of the value function $\Delta V_i^{\pi}(\bar{\varepsilon}_{ik})$ as

$$\Delta V_i^{\pi}(\bar{\varepsilon}_{ik}) = V_i^{\pi}(\bar{\varepsilon}_{i(k+1)}) - V_i^{\pi}(\bar{\varepsilon}_{ik}) \tag{18}$$

and its gradient as

$$\nabla V_i^{\pi}(\bar{\varepsilon}_{i(k+1)}) = \frac{\partial V_i^{\pi}(\bar{\varepsilon}_{i(k+1)})}{\partial \bar{\varepsilon}_{i(k+1)}}. \tag{19}$$

The objective of the graphical games optimization problem is to find for each i the optimal value function $V_i^{o}(\bar{\varepsilon}_{ik})$ such that

$$V_i^{\mathrm{o}}(\bar{\varepsilon}_{ik}) = \min_{u_i}(V_i(\bar{\varepsilon}_{ik})) = \min_{u_i}(\sum_{l=k}^{\infty} U_i(\varepsilon_{il}, u_{il}, u_{-il})). \tag{20}$$

Given stationary admissible polices for the neighbors of agent i. Applying the Bellman optimality principle yields

$$V_i^{\mathrm{o}}(\bar{\varepsilon}_{ik}) = \min_{u_{ik}}(U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) + V_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)})). \tag{21}$$

Consequently, the optimal control policy for each agent i is given by

$$u_{ik}^{\mathrm{o}} = \arg\min_{u_{ik}}(U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) + V_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)})). \tag{22}$$

Then,

$$\begin{aligned} \pi_{ik} &= u_{ik}^{\mathrm{o}} \\ &= R_{ii}^{-1}([\cdots\ (g_i + d_i)\ \cdots\ -e_{ji}\ \cdots] \otimes B_i^{\mathrm{T}}) \nabla V_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}) \\ &= M_i \nabla V_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}). \end{aligned} \tag{23}$$

Substituting (23) into (21) yields the graphical game Bellman optimality equations

$$\begin{aligned} &V_i^{\mathrm{o}}(\bar{\varepsilon}_{ik}) \\ &= \frac{1}{2}(\varepsilon_{ik}^{\mathrm{T}} Q_{ii} \varepsilon_{ik} + \nabla V_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)})^{\mathrm{T}} M_i^{\mathrm{T}} R_{ii} M_i \nabla V_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}) \\ &\quad + \sum_{j \in N_i} \nabla V_j^{\mathrm{o}}(\bar{\varepsilon}_{j(k+1)})^{\mathrm{T}} M_j^{\mathrm{T}} R_{ij} M_j \nabla V_j^{\mathrm{o}}(\bar{\varepsilon}_{j(k+1)})) \\ &\quad + V_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}) \end{aligned} \tag{24}$$

with initial conditions $V_i^{\mathrm{o}}(\mathbf{0}) = 0, \forall i$.

3.4 Q-function based Bellman equations

Herein, ADHDP structure for single agent is extended to formulate and solve the dynamic graphical game without knowing any of the agents' dynamics where only local measurements are used. The solution for the graphical game is given in terms of the solution to a set of coupled DTHJB equations.

The Q-function for each agent i is defined as follows:

$$Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik}) = U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) + V_i^{\pi}(\bar{\varepsilon}_{i(k+1)}). \tag{25}$$

Since the policies $\pi_{jk}, j \in N_i$ are stationary admissible, this is in fact a best response Bellman equation. Note that $Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik})$ is defined such that

$$Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik}) = V_i^{\pi}(\bar{\varepsilon}_{ik}). \tag{26}$$

Therefore, the best response Bellman equation is given by

$$Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik}) = U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) + Q_i^{\pi}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) \tag{27}$$

with initial conditions $Q_i^{\pi}(\mathbf{0}) = 0, \forall i$.

Define the difference of the Q-function $Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik})$ as

$$\Delta Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik}) = Q_i^{\pi}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) - Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik}) \tag{28}$$

and its gradient as

$$\nabla Q_i^{\pi}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) = \frac{\partial Q_i^{\pi}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})}{\partial \bar{\varepsilon}_{i(k+1)}}. \tag{29}$$

The optimal control policy for each agent i is given such that

$$\pi_{ik} = \tilde{u}_{ik}^{\mathrm{o}} = \arg\min_{u_{ik}}(Q_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik})). \tag{30}$$

Then,

$$R_{ii} \tilde{u}_{ik}^{\mathrm{o}} + \frac{\partial Q_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})}{\partial u_{ik}} = 0. \tag{31}$$

Therefore, the optimal control policy is given by

$$\begin{aligned} \tilde{u}_{ik}^{\mathrm{o}} =& (g_i + d_i) R_{ii}^{-1} B_i^{\mathrm{T}} \frac{\partial Q_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})}{\partial \varepsilon_{i(k+1)}} \\ &- R_{ii}^{-1}([\cdots\ e_{ji}\ \cdots] \otimes B_i^{\mathrm{T}}) \frac{\partial Q_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})}{\partial \varepsilon_{-i(k+1)}}. \end{aligned} \tag{32}$$

Rearranging this equation yields

$$\begin{aligned} \pi_{ik} = \tilde{u}_{ik}^{\mathrm{o}} =& R_{ii}^{-1}([\cdots\ (g_i + d_i) \cdots\ -e_{ji}\ \cdots] \\ &\otimes B_i^{\mathrm{T}}) \nabla Q_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), \end{aligned} \tag{33}$$

which is the same as (23). Then,

$$\pi_{ik} = \tilde{u}^{\mathrm{o}}_{ik} = M_i \nabla Q^{\mathrm{o}}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}). \tag{34}$$

Thus, the best response Bellman optimality equation based on the Q-function (25) and optimal policies (34) is given by

$$\begin{aligned} Q^{\mathrm{o}}_i(\bar{\varepsilon}_{ik}, \tilde{u}^{\mathrm{o}}_{ik}) = \frac{1}{2}(\varepsilon^{\mathrm{T}}_{ik} Q_{ii} \varepsilon_{ik} + \tilde{u}^{o\mathrm{T}}_{ik} R_{ii} \tilde{u}^{\mathrm{o}}_{ik} + \sum_{j \in N_i} \tilde{u}^{o\mathrm{T}}_{jk} R_{ij} \tilde{u}^{\mathrm{o}}_{jk}) \\ + Q^{\mathrm{o}}_i(\bar{\varepsilon}_{i(k+1)}, \tilde{u}^{\mathrm{o}}_{i(k+1)}), \end{aligned} \tag{35}$$

or

$$\begin{aligned} & Q^{\mathrm{o}}_i(\bar{\varepsilon}_{ik}, \tilde{u}^{\mathrm{o}}_{ik}) \\ & = Q^{\mathrm{o}}_i(\bar{\varepsilon}_{i(k+1)}, \tilde{u}^{\mathrm{o}}_{i(k+1)}) + \frac{1}{2}(\varepsilon^{\mathrm{T}}_{ik} Q_{ii} \varepsilon_{ik} \\ & \quad + \nabla Q^{o\mathrm{T}}_i(\bar{\varepsilon}_{i(k+1)}, \tilde{u}^{\mathrm{o}}_{i(k+1)}) M^{\mathrm{T}}_i R_{ii} M_i \nabla Q^{\mathrm{o}}_i(\bar{\varepsilon}_{i(k+1)}, \tilde{u}^{\mathrm{o}}_{i(k+1)}) \\ & \quad + \sum_{j \in N_i} \nabla Q^{o\mathrm{T}}_j(\bar{\varepsilon}_{j(k+1)}, \tilde{u}^{\mathrm{o}}_{j(k+1)}) M^{\mathrm{T}}_j R_{ij} M_j \nabla Q^{\mathrm{o}}_j(\bar{\varepsilon}_{j(k+1)}, \tilde{u}^{\mathrm{o}}_{j(k+1)})), \end{aligned} \tag{36}$$

which is equivalent to (24).

3.5 Coupled Hamilton-Jacobi-Bellman equations

The Hamilton-Jacobi theory [9] is used to relate the Hamiltonian functions (13) and the Bellman equations (27).

Considering the Hamiltonian (13) and the Q-function $Q^{\pi}_i(\bar{\varepsilon}_{ik}, u_{ik})$ given by (26) or (27). Then, $Q^{\pi}_i(\bar{\varepsilon}_{ik}, u_{ik})$ satisfies the Discrete-Time Hamilton Jacobi (DTHJ) equation

$$\begin{aligned} & \Delta Q^{\pi}_i(\bar{\varepsilon}_{ik}, u_{ik}) - \nabla Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)} \\ & + H^{\pi}_i(\bar{\varepsilon}_{ik}, \nabla Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u_{ik}, \pi_{-ik}) = 0. \end{aligned} \tag{37}$$

This equation provides the motivation for defining the costate variable $\lambda_{i(k+1)}$ in terms of the Q-function such that

$$\lambda_{i(k+1)} = \nabla Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}). \tag{38}$$

The optimal control policy based on the Bellman optimality equation (27) is given by (34). The next result relates the Hamiltonian (13) along the optimal trajectories and the Bellman optimality equation (36).

Theorem 1 (Discrete-time coupled HJB equation)

a) Let $0 < Q^{\pi*}_i(\bar{\varepsilon}_{ik}, u^*_{ik}) \in \mathbb{C}^2, \forall i$ satisfy the coupled DTHJB equation

$$\begin{aligned} & H^{\pi}_i(\bar{\varepsilon}_{ik}, \nabla Q^{\pi*}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u^*_{ik}, u^*_{-ik}) \\ & = \nabla Q^{\pi*}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)} + U_i(\varepsilon_{ik}, u^*_{ik}, u^*_{-ik}) = 0 \end{aligned} \tag{39}$$

with initial condition $Q^*_i(\mathbf{0}) = 0$, where

$$\pi_{ik} = u^*_{ik} = M_i \nabla Q^{\pi*}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}). \tag{40}$$

Then, $Q^{\pi*}_i(\bar{\varepsilon}_{ik}, u^*_{ik})$ satisfies the Bellman optimality equation (36).

b) Let $(A, B_i), \forall i$ be reachable for each agent i. Let $0 < Q^{\pi*}_i(\bar{\varepsilon}_{ik}, u^*_{ik}) \in \mathbb{C}^2, \forall i$ satisfy (36). Then $Q^{\pi*}_i(\bar{\varepsilon}_{ik}, u^*_{ik})$ satisfies (39).

Proof a) If $Q^{\pi*}_i(\bar{\varepsilon}_{ik}, u^*_{ik})$ satisfies (39) and u^*_{ik} is given by (40), then $H^{\pi}_i(\bar{\varepsilon}_{ik}, \nabla Q^{\pi*}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u^*_{ik}, u^*_{-ik}) = 0$. Then, the DTHJ equation yields $\Delta Q^{\pi*}_i(\bar{\varepsilon}_{ik}, u^*_{ik}) = \nabla Q^{\pi*}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)}$. Therefore, $Q^{\pi*}_i(\bar{\varepsilon}_{ik}, u^*_{ik})$ satisfies (36).

b) Completing the squares on the Hamiltonian (13) for arbitrary smooth function $Q^{\pi}_i(\bar{\varepsilon}_{ik}, u_{ik})$ yields

$$\begin{aligned} & H^{\pi}_i(\varepsilon_i, \nabla Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u_{ik}, \pi_{-ik}) \\ & = H^{\pi}_i(\varepsilon_i, \nabla Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u^*_{ik}, u^*_{-ik}) \\ & \quad + \frac{1}{2}(u_{ik} - u^*_{ik})^{\mathrm{T}} R_{ii}(u_{ik} - u^*_{ik}) \\ & \quad + \sum_{j \in N_i} u^{*\mathrm{T}}_{jk} R_{ij}(\pi_{jk} - u^*_{jk}) \\ & \quad + \frac{1}{2} \sum_{j \in N_i} (\pi_{jk} - u^*_{jk})^{\mathrm{T}} R_{ij}(\pi_{jk} - u^*_{jk}) \\ & \quad + \nabla_{-i} Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} (\bar{g}_i \bar{B}_i(\bar{u}_{-ik} - \bar{u}^*_{-ik})) \\ & \quad + \nabla_i Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \sum_{j \in N_i} e_{ij} B_j(\pi_{jk} - u^*_{jk}), \end{aligned} \tag{41}$$

where

$$\begin{aligned} & \bar{u}_{-ik} = [u^{\mathrm{T}}_j \;\; u^{\mathrm{T}}_c \;\; u^{\mathrm{T}}_{c'} \;\; u^{\mathrm{T}}_{j'}]^{\mathrm{T}}, \;\; \{j, j'\} \in N_i, \;\; c \in N_j, \\ & c' \in N_{j'}, \{c, c'\} \neq i, \;\; u^*_{ik} = M_i \nabla Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), \\ & \bar{B}_i = \mathrm{diag}\{B_j, B_c, B_{c'}, \ldots, B_{j'}\}, \\ & \bar{g}_i = \begin{bmatrix} -(g_j + d_j) & e_{jc} & e_{jc'} & e_{jj'} \\ \vdots & \vdots & \vdots & \vdots \\ e_{j'j} & e_{j'c} & e_{j'c'} & -(g_{j'} + d_{j'}) \end{bmatrix} \otimes I_n, \\ & \nabla_i Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) = \frac{\partial Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})}{\partial \varepsilon_{i(k+1)}}, \\ & \nabla_{-i} Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) = \frac{\partial Q^{\pi}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})}{\partial \varepsilon_{-i(k+1)}}. \end{aligned}$$

The Hamiltonian with smooth value function $Q^{\pi*}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})$ for arbitrary policy u_{ik} is given by

$$H^{\pi}_i(\varepsilon_i, \nabla Q^{\pi*}_i(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u_{ik}, \pi_{-ik})$$

$$
\begin{aligned}
&= U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) + (\nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)}) \\
&= \frac{1}{2}(u_{ik} - u_{ik}^*)^{\mathrm{T}} R_{ii}(u_{ik} - u_{ik}^*) + \sum_{j\in N_i} u_{jk}^{*\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
&\quad + \frac{1}{2}\sum_{j\in N_i}(\pi_{jk} - u_{jk}^*)^{\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
&\quad + \nabla_{-i} Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} (\bar{g}_i \bar{B}_i(\bar{u}_{-ik} - \bar{u}_{-ik}^*)) \\
&\quad + \nabla_i Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \sum_{j\in N_i} e_{ij} B_j(\pi_{jk} - u_{jk}^*). \qquad (42)
\end{aligned}
$$

Let the value function $\tilde{Q}_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik})$ satisfy Bellman equation (27). Rearranging this equation yields

$$
\begin{aligned}
\tilde{Q}_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik}) = {} & U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) \\
& + \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)} \\
& - \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)} \\
& + \tilde{Q}_i^{\pi}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}). \qquad (43)
\end{aligned}
$$

Introducing the Hamiltonian (42) into this equation yields

$$
\begin{aligned}
\tilde{Q}_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik}) = {} & \frac{1}{2}(u_{ik} - u_{ik}^*)^{\mathrm{T}} R_{ii}(u_{ik} - u_{ik}^*) \\
& + \frac{1}{2}\sum_{j\in N_i}(\pi_{jk} - u_{jk}^*)^{\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
& + \sum_{j\in N_i} u_{jk}^{*\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
& + \nabla_{-i} Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} (\bar{g}_i \bar{B}_i(\bar{u}_{-ik} - \bar{u}_{-ik}^*)) \\
& + \nabla_i Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \sum_{j\in N_i} e_{ij} B_j(\pi_{jk} - u_{jk}^*) \\
& - (\nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)}) \\
& + \tilde{Q}_i^{\pi}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}). \qquad (44)
\end{aligned}
$$

Applying the Bellman's optimality principle, yields that $\tilde{Q}_i^{\mathrm{o}}(\bar{\varepsilon}_{ik}, u_{ik})$ satisfies the following optimality equation

$$
\begin{aligned}
\tilde{Q}_i^{\mathrm{o}}(\bar{\varepsilon}_{ik}, u_{ik}^{\mathrm{o}}) = {} & \min_{u_{ik}}(\tilde{Q}_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik})) \\
= {} & \min_{u_{ik}}(\tfrac{1}{2}(u_{ik} - u_{ik}^*)^{\mathrm{T}} R_{ii}(u_{ik} - u_{ik}^*) \\
& + \sum_{j\in N_i} u_{jk}^{*\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
& + \frac{1}{2}\sum_{j\in N_i}(\pi_{jk} - u_{jk}^*)^{\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
& + \nabla_{-i} Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} (\bar{g}_i \bar{B}_i(\bar{u}_{-ik} - \bar{u}_{-ik}^*)) \\
& + \nabla_i Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \sum_{j\in N_i} e_{ij} B_j(\pi_{jk} - u_{jk}^*) \\
& - (\nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)}) \\
& + \tilde{Q}_i^{\pi}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})). \qquad (45)
\end{aligned}
$$

Applying the stationarity condition $\dfrac{\partial \tilde{Q}_i^{\pi}(\bar{\varepsilon}_{ik}, u_{ik})}{\partial u_{ik}} = 0$ yields the control policy u_{ik}^{o} such that

$$
\begin{aligned}
& R_{ii} M_i(\nabla \tilde{Q}_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) - \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})) \\
& + R_{ii}(u_{ik}^{\mathrm{o}} - u_{ik}^*) = 0.
\end{aligned}
$$

Then,

$$
\begin{aligned}
(u_{ik}^{\mathrm{o}} - u_{ik}^*) = {} & M_i(\nabla \tilde{Q}_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) \\
& - \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})). \qquad (46)
\end{aligned}
$$

The hessians of the Hamiltonians (13) and Bellman equations (27) are positive definite such that $\nabla^2_{u_{ik}}(H_i^{\pi}) = R_{ii} > 0$ and $\nabla^2_{u_{ik}}(\tilde{Q}_i^{\pi}) = R_{ii} > 0$. Therefore, the optimal control policy is unique, that is $u_{ik}^* = u_{ik}^{\mathrm{o}}, \forall k$.

Equations (46) and (16) yield

$$
\begin{cases}
(g_i + d_i) R_{ii}^{-1} B_i^{\mathrm{T}} A^{\mathrm{T}p}(\nabla_i \tilde{Q}_i^{\mathrm{o}} - \nabla_i Q_i^{\pi*}) = 0, \\
e_{ji} R_{ii}^{-1} B_i^{\mathrm{T}} A^{\mathrm{T}p}(\nabla_j \tilde{Q}_i^{\mathrm{o}} - \nabla_j Q_i^{\pi*}) = 0, \quad p = 0, 1, \ldots, n-1.
\end{cases} \qquad (47)
$$

The reachability matrix

$$
\tilde{U}_i = [B_i \;\; AB_i \;\; A^2 B_i \;\; \cdots \;\; A^{n-1} B_i] \qquad (48)
$$

under the hypothesis that the reachability matrix $\tilde{U}_i$ has full rank. Therefore, since $Q_i^{\pi*}(\mathbf{0}) = 0$ and $\tilde{Q}_i^{\mathrm{o}}(\mathbf{0}) = 0$ then,

$$
\tilde{Q}_i^{\mathrm{o}}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}) = Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), \quad \forall k. \qquad (49)
$$

□

Associated to the optimal value functions $Q_i^{\pi*}(\cdot)$, define the optimal performance indices (8) as $J_i^*(\cdot)$ with optimal control policies given by (40).

4 Nash solution for the dynamic graphical game

The objective of the dynamic graphical game is to solve the non-cooperative minimization problems (20), which lead to the Bellman optimality equations (36). In this section, it is shown that, the solution of the coupled Bellman optimality equations (36) is a Nash equilibrium solution for the dynamic graphical game.

4.1 Nash equilibrium for the graphical games

Nash equilibrium solution for the game is defined as follows [23]

Definition 3 The N-player game with N-tuple of optimal control policies $\{u_1^*, u_2^*, \ldots, u_N^*\}$ is said to have a

Nash equilibrium solution if for all $i \in N$

$$J_i^* \triangleq J_i(u_i^*, u_{\bar{i}}^*) \leqslant J_i(u_i, u_{\bar{i}}^*). \tag{50}$$

4.2 Stability and Nash solution for the graphical games

Stable Nash solution for the dynamic graphical game is shown to be equivalent to the solution of the underlying coupled Bellman optimality equations (24) or (36).

Theorem 2 (Stability and Nash equilibrium solution) Let $0 < Q_i^{\pi*}(\bar{\varepsilon}_{ik}, u_{ik}^*) \in \mathbb{C}^2$ satisfy the DTHJB (39) or the Bellman optimality equation (36). Let all agents use the control policies (40). Then,

a) all agents synchronize to the leader's dynamics (2);

b) the optimal performance index is $J_{il}^* = Q_i^{\pi*}(\bar{\varepsilon}_{il}, u_{il}^*)$; and

c) the tuple $\{u_i^*, u_{\bar{i}}^*\}$ represents the Nash equilibrium solution for the graphical game.

Proof a) The value function $Q_i^{\pi*}(\bar{\varepsilon}_{ik}, u_{ik}^*)$ satisfies the Bellman optimality equation (36) such that

$$Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, u_{i(k+1)}^*) - Q_i^{\pi*}(\bar{\varepsilon}_{ik}, u_{ik}^*) = -U_i^*(\varepsilon_{ik}, u_{ik}^*, u_{-ik}^*) < 0. \tag{51}$$

Therefore, $Q_i^{\pi*}(\bar{\varepsilon}_{ik}, u_{ik}^*)$ serves as a Lyapunov function, and the error systems (7) are asymptotically stable. Let the graph have a spanning tree (i.e., the graph is strongly connected) with the pinning gain into at least one root agent nonzero. Then according to (6), all agents asymptotically synchronize to the leader's dynamics (2).

b) Using Theorem 1 and DTHJB (39), then the Hamiltonian (41) for arbitrary control policies is given by

$$\begin{aligned}
&H_i^{\pi}(\varepsilon_{ik}, \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u_{ik}, \pi_{-ik}) \\
&= H_i^{\pi}(\varepsilon_i, \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}^*, \pi_{i(k+1)}), u_{ik}^*, u_{jk}^*) \\
&\quad + \frac{1}{2}(u_{ik} - u_{ik}^*)^{\mathrm{T}} R_{ii}(u_{ik} - u_{ik}^*) + \sum_{j \in N_i} u_{jk}^{*\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
&\quad + \frac{1}{2} \sum_{j \in N_i} (\pi_{jk} - u_{jk}^*)^{\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
&\quad + \nabla_{-i} Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} (\bar{g}_i \bar{B}_i(\bar{\pi}_{-ik} - \bar{u}_{-ik}^*)) \\
&\quad + \nabla_i Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \sum_{j \in N_i} e_{ij} B_j(\pi_{jk} - u_{jk}^*).
\end{aligned} \tag{52}$$

The performance index at time index l is given by

$$J_{il}^{\pi} = Q_i^{\pi*}(\bar{\varepsilon}_i(\infty), \pi_i(\infty)) + \sum_{k=l}^{\infty} U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}). \tag{53}$$

The result in part a) yields, $Q_i^{\pi*}(\bar{\varepsilon}_i(\infty), \pi_i(\infty)) = 0$. Rearranging (53) yields

$$\begin{aligned}
J_{il}^{\pi} &= Q_{il}^{\pi*}(\bar{\varepsilon}_{il}, u_{il}^*) \\
&\quad + \sum_{k=l}^{\infty} (U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) - U_i^*(\varepsilon_{ik}, u_{ik}^*, u_{-ik}^*)).
\end{aligned} \tag{54}$$

The Hamiltonian for arbitrary control inputs is given by

$$\begin{aligned}
&H_i^{\pi}(\varepsilon_{ik}, \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)}), u_{ik}, \pi_{-ik}) \\
&= U_i(\varepsilon_{ik}, u_{ik}, \pi_{-ik}) + (\nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)})|_{u_{ik}, \pi_{-ik}}.
\end{aligned} \tag{55}$$

The Hamiltonian for optimal control inputs is given by

$$\begin{aligned}
&H_i^{\pi}(\varepsilon_{ik}, \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}^*, \pi_{i(k+1)}), u_{ik}^*, u_{-ik}^*) \\
&= \nabla Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \bar{\varepsilon}_{i(k+1)}|_{u_{ik}^*, u_{-ik}^*} + U_i^*(\varepsilon_{ik}, u_{ik}^*, u_{-ik}^*) \\
&= 0.
\end{aligned} \tag{56}$$

Using (55) and (56) in (54) yields

$$\begin{aligned}
J_{il}^{\pi} &= Q_{il}^{\pi*}(\bar{\varepsilon}_{il}, u_{il}^*) + \sum_{k=l}^{\infty} (\frac{1}{2}(u_{ik} - u_{ik}^*)^{\mathrm{T}} R_{ii}(u_{ik} - u_{ik}^*) \\
&\quad + \frac{1}{2} \sum_{j \in N_i} (\pi_{jk} - u_{jk}^*)^{\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
&\quad + \sum_{j \in N_i} u_{jk}^{*\mathrm{T}} R_{ij}(\pi_{jk} - u_{jk}^*) \\
&\quad + \nabla_{-i} Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} (\bar{g}_i \bar{B}_i(\bar{\pi}_{-ik} - \bar{u}_{-ik}^*)) \\
&\quad + \nabla_i Q_i^{\pi*}(\bar{\varepsilon}_{i(k+1)}, \pi_{i(k+1)})^{\mathrm{T}} \sum_{j \in N_i} e_{ij} B_j(\pi_{jk} - u_{jk}^*)).
\end{aligned} \tag{57}$$

At optimal control policies ($u_{ik} = u_{ik}^*, \pi_{jk} = u_{jk}^*$), the optimal performance index (at time index l) (57) is given by the unique value $Q_{il}^{\pi*}(\bar{\varepsilon}_{il}, u_{il}^*)$ such that

$$J_{il}^* = Q_{il}^{\pi*}(\bar{\varepsilon}_{il}, u_{il}^*). \tag{58}$$

c) Given that the summation of the performance index (54) is always positive for arbitrary control policies such that

$$\sum_{k=l}^{\infty} U_i(\varepsilon_{ik}, u_{ik}, u_{-ik}^*) - U_i^*(\varepsilon_{ik}, u_{ik}^*, u_{-ik}^*) > 0. \tag{59}$$

Then, according to (59), the argument of the performance index (57) is positive for arbitrary control policy. Then, (59) and (54) yield

$$J_{il}^*(u_{il}^*, u_{\bar{i}l}^*) \leqslant J_{il}^{\pi}(u_{il}, u_{\bar{i}l}^*). \tag{60}$$

Therefore, Nash equilibrium exists according to Definition 3. □

5 Model-free reinforcement learning solution

In this section, an online model-free policy iteration algorithm is developed to solve the discrete-time dynamic graphical games in real-time. This is a cooperative version of the ADP single agent's methods introduced in [1, 2]. Specifically, the single agent (ADHDP) algorithm is extended to solve the multi-player dynamic graphical game.

The Q-function is expressed in terms of the agent control input, state of each agent i, and the states of its neighbors such that

$$\tilde{Q}_i = \frac{1}{2}[\,\bar{\varepsilon}_{ik}^{\mathrm{T}} \;\; u_{ik}^{\mathrm{T}}]\tilde{H}_i \begin{bmatrix} \bar{\varepsilon}_{ik} \\ u_{ik} \end{bmatrix}, \tag{61}$$

$$\tilde{H}_i = \begin{bmatrix} \tilde{H}_{i(\varepsilon_{ik}\varepsilon_{ik})} & \tilde{H}_{i(\varepsilon_{ik}\varepsilon_{yk})} & \cdots & \tilde{H}_{i(\varepsilon_{ik}\varepsilon_{rk})} & \tilde{H}_{i(\varepsilon_{ik}u_{ik})} \\ \tilde{H}_{i(\varepsilon_{yk}\varepsilon_{ik})} & \tilde{H}_{i(\varepsilon_{yk}\varepsilon_{yk})} & \cdots & \tilde{H}_{i(\varepsilon_{yk}\varepsilon_{rk})} & \tilde{H}_{i(\varepsilon_{yk}u_{ik})} \\ \vdots & \vdots & & \vdots & \vdots \\ \tilde{H}_{i(\varepsilon_{rk}\varepsilon_{ik})} & \tilde{H}_{i(\varepsilon_{rk}\varepsilon_{yk})} & \cdots & \tilde{H}_{i(\varepsilon_{rk}\varepsilon_{rk})} & \tilde{H}_{i(\varepsilon_{rk}u_{ik})} \\ \tilde{H}_{i(u_{ik}\varepsilon_{ik})} & \tilde{H}_{i(u_{ik}\varepsilon_{yk})} & \cdots & \tilde{H}_{i(u_{ik}\varepsilon_{rk})} & \tilde{H}_{i(u_{ik}u_{ik})} \end{bmatrix},$$

where $\tilde{H}_i$ is the solution matrix for each agent i, $\tilde{H}_{i(u_{ik}\varepsilon_{ik})}$ is a sub-block of the matrix $\tilde{H}_i$ with the appropriate position indices $(u_{ik}, \varepsilon_{ik})$. The optimal control policy is given so that

$$\begin{aligned} &\frac{\partial \tilde{Q}_i}{\partial u_{ik}} = \arg\min_{u_{ik}}(\tilde{Q}_i) = 0 \\ &\Rightarrow 2\tilde{H}_{i(u_{ik}u_{ik})}u_{ik} + 2\tilde{H}_{i(u_{ik}\varepsilon_{ik})}\varepsilon_{ik} + \sum_{j\in N_i} 2\tilde{H}_{i(u_{ik}\varepsilon_{jk})}\varepsilon_{jk} = 0. \end{aligned} \tag{62}$$

Then,

$$u_{ik} = -\tilde{H}^{-1}_{i(u_{ik}u_{ik})}\tilde{H}_{i(u_{ik}\varepsilon_{ik})}\varepsilon_{ik} - \sum_{j\in N_i} \tilde{H}^{-1}_{i(u_{ik}u_{ik})}\tilde{H}_{i(u_{ik}\varepsilon_{jk})}\varepsilon_{jk}. \tag{63}$$

The following algorithm solves the Bellman optimality equations (36) for the optimal game values and policies.

Algorithm 1 (Model-free policy iteration algorithm)

1) Start with arbitrary initial admissible policies u_{ik}^0 and values $\tilde{Q}_i^0(\bar{\varepsilon}_{ik}, u_{ik}^0)$.

2) Solve for $\tilde{Q}_i^l(\bar{\varepsilon}_{ik}, u_{ik}^l), \forall i$ using

$$\tilde{Q}_i^l(\bar{\varepsilon}_{ik}, u_{ik}^l) = U_i(\varepsilon_{ik}, u_{ik}^l, u_{-ik}^l) + \tilde{Q}_i^l(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^l, u_{-ik}^l)}, u_{i(k+1)}^l). \tag{64}$$

3) Find the policy u_{ik}^{l+1} using

$$u_{ik}^{l+1} = -\tilde{H}^{-1(l)}_{i(u_{ik}u_{ik})}\tilde{H}^{l}_{i(u_{ik}\varepsilon_{ik})}\varepsilon_{ik} - \sum_{j\in N_i} \tilde{H}^{-1(l)}_{i(u_{ik}u_{ik})}\tilde{H}^{l}_{i(u_{ik}\varepsilon_{jk})}\varepsilon_{jk}, \;\forall i. \tag{65}$$

4) On convergence of $\|\tilde{H}_i^{l+1} - \tilde{H}_i^l\|$ End, otherwise repeat steps 2) and 3).

Remark 2 If Algorithm 1 converges to the optimal value function $Q_i^{\pi*}(\bar{\varepsilon}_{ik}, u_{ik}^*), \forall i$ and the optimal policies $u_{ik}^*, \forall i$ then (64) and (65) hold simultaneously. In view of (63) and Theorem 1, the value functions $Q_i^{\pi*}(\bar{\varepsilon}_{ik}, u_{ik}^*), \forall i$ satisfy the coupled Bellman optimality equations (36) and the coupled DTHJB equations (39).

Remark 3 This algorithm does not require the knowledge of any of the agents' dynamics in the systems (7).

Remark 4 The Policy improvement step (65) in Algorithm 1 does not require the graph to be undirected as previously imposed by the optimal policy structure (34) where the out-neighbor information are required. Thus, step (65) enables Algorithm 1 to solve the dynamic graphical games with directed or undirected graph topologies.

Remark 5 To solve the Bellman equations (64), numerous instances of (64) must be obtained at successive time instants. For these equations to be independent, a persistence of excitation condition is needed as per standard usage. This is further discussed in Remark 6 immediately before Section 6.1.

The following theorem provides the convergence proof for Algorithm 1 when all agents update their policies simultaneously.

Theorem 3 (Policy iteration convergence proof) Let all agents perform Algorithm 1 simultaneously. Assume that all the initial policies $u_{ik}^0, \forall i$ are admissible. Suppose that $\bar{\sigma}(R_{jj}^{-1}R_{ij}), \forall i$ are small. Then,

a) $u_{ik}^l, \forall i, l > 0$ are stabilizing and hence admissible policies.

b) Policy iteration Algorithm 1 generates monotonically decreasing sequence of value functions $\tilde{Q}_{ik}^l, \forall i$ such that $0 \leqslant \ldots \leqslant \tilde{Q}_{ik}^{l+1} \leqslant \tilde{Q}_{ik}^l, \forall i$ and this sequence converges to the best response value functions $\tilde{Q}_i^*, \forall i$ that satisfy the coupled DTHJB equations (39).

Proof a) Equations (27) or (64) yield,

$$\tilde{Q}_i^l(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^l, u_{-ik}^l)}, u_{i(k+1)}^l) - \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^l, u_{-ik}^l)}, u_{ik}^l) < 0, \;\forall i, l. \tag{66}$$

Thus, the value functions $\tilde{Q}_i^l, \forall i, l$ are Lyapunov functions for arbitrary policies $u_i^l, \forall i, l$.

Assuming $u_{ik}^0, \forall i$ are admissible, then there exist value functions $\tilde{Q}_i^l, \forall i$ that satisfy (26) or (64) such that

$$\begin{aligned}\tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^l, u_{-ik}^l)}, u_{ik}^l) &= \sum_{m=k}^{\infty} U_i(\varepsilon_{im}, u_{im}^l, u_{-im}^l) \\ &= \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^l)}, u_{ik}^{l+1}) + \Delta\tilde{U}_k(u_i^l, u_i^{l+1}),\end{aligned} \tag{67}$$

where $\Delta\tilde{U}_k(u_i^l, u_i^{l+1}) = \sum_{m=k}^{\infty} (\frac{1}{2}(u_{im}^l - u_{im}^{l+1})^{\mathrm{T}} R_{ii}(u_{im}^l - u_{im}^{l+1}) + u_{im}^{(l+1)\mathrm{T}} R_{ii}(u_{im}^l - u_{im}^{l+1}))$.

The policies $u_{ik}^{l+1}, \forall i, l$ are given by (65). Therefore, $\Delta\tilde{U}_k(u_i^l, u_i^{l+1}) > 0$ and (67) yields

$$\tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^l, u_{-ik}^l)}, u_{ik}^l) > \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^l)}, u_{ik}^{l+1}). \tag{68}$$

Similarly,

$$\begin{aligned}&\tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^l)}, u_{ik}^{l+1}) \\ &= \sum_{m=k}^{\infty} U_i(\varepsilon_{im}, u_{im}^{l+1}, u_{-im}^l) \\ &= \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1}) + \Delta\tilde{U}_k(u_{-i}^l, u_{-i}^{l+1}).\end{aligned} \tag{69}$$

The assumption that $\Delta\tilde{U}_k(u_{-i}^l, u_{-i}^{l+1}) > 0$, guarantees that

$$\tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^l)}, u_{ik}^{l+1}) > \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1}). \tag{70}$$

Thus $\Delta\tilde{U}_k(u_{-i}^l, u_{-i}^{l+1}) \geqslant 0$ is a sufficient condition for stabilization is, which is guaranteed by

$$\begin{aligned}&\sum_{j\in N_i} \frac{1}{2}(u_{jk}^l - u_{jk}^{l+1})^{\mathrm{T}} R_{ij}(u_{jk}^l - u_{jk}^{l+1}) \\ &- u_{jk}^{(l+1)\mathrm{T}} R_{ij}(u_{jk}^{l+1} - u_{jk}^l) > 0.\end{aligned} \tag{71}$$

Using the norm properties on this inequality yields

$$\begin{aligned}&\sum_{j\in N_i} \frac{1}{2}\underline{\sigma}(R_{ij})\|\Delta u_{jk}^l\| \\ &> \sum_{j\in N_i} (g_j + d_j)\bar{\sigma}(R_{jj}^{-1}R_{ij})((g_j + d_j)\|\nabla_j \tilde{V}_j^l(\bar{\varepsilon}_{j(k+1)}^{u_{jk}^l, u_{-jk}^l})\| \\ &\quad + \sum_{o\in N_j} (e_{jo}\|\nabla_0 \tilde{V}_j^l(\bar{\varepsilon}_{j(k+1)}^{u_{jk}^l, u_{-jk}^l})\|))\|B_{jk}\|,\end{aligned} \tag{72}$$

where $\Delta u_{jk}^l = (u_{jk}^l - u_{jk}^{l+1})$.

Under the assumption (72). Inequalities (68) and (70) yield

$$\tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^l, u_{-ik}^l)}, u_{ik}^l) > \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^l)}, u_{ik}^{l+1}) > \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1}). \tag{73}$$

The assumption $\Delta\tilde{U}_k(u_{-i}^l, u_{-i}^{l+1}) \geqslant 0$ indicates that choosing small values of $\bar{\sigma}(R_{jj}^{-1}R_{ij})$ guarantees that (71) is satisfied. This can be guaranteed for any choice of R_{ij} of agent i by selecting R_{jj} large enough. Therefore, $u_{ik}^{l+1}, \forall i, l$ are stabilizing policies and hence admissible.

b) Equation (66) yields

$$\tilde{Q}_i^l(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{i(k+1)}^{l+1}) - \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1}) < 0. \tag{74}$$

Equation (26) or (64) yields

$$\begin{aligned}&\tilde{Q}_i^{l+1}(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{i(k+1)}^{l+1}) - \tilde{Q}_i^{l+1}(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1}) \\ &+ U_i(\varepsilon_{ik}, u_{ik}^{l+1}, u_{-ik}^{l+1}) = 0.\end{aligned} \tag{75}$$

Equations (74), (75), and the assumption (72) yield

$$\begin{aligned}&\tilde{Q}_i^l(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{i(k+1)}^{l+1}) - \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1}) \\ &\leqslant \tilde{Q}_i^{l+1}(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{i(k+1)}^{l+1}) - \tilde{Q}_i^{l+1}(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1}).\end{aligned} \tag{76}$$

Applying the summation on (76) such that

$$\begin{aligned}&\sum_{k=\bar{K}}^{\infty} (\tilde{Q}_i^l(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{i(k+1)}^{l+1}) - \tilde{Q}_i^l(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1})) \\ &< \sum_{k=\bar{K}}^{\infty} (\tilde{Q}_i^{l+1}(\bar{\varepsilon}_{i(k+1)}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{i(k+1)}^{l+1}) - \tilde{Q}_i^{l+1}(\bar{\varepsilon}_{ik}^{(u_{ik}^{l+1}, u_{-ik}^{l+1})}, u_{ik}^{l+1})).\end{aligned}$$

This reduces to

$$\begin{aligned}&\tilde{Q}_i^l(\bar{\varepsilon}_{i(\infty)}^{(u_{i\infty}^{l+1}, u_{-i\infty}^{l+1})}, u_{i\infty}^{l+1}) - \tilde{Q}_i^l(\bar{\varepsilon}_{i\bar{K}}^{(u_{i\bar{K}}^{l+1}, u_{-i\bar{K}}^{l+1})}, u_{i\bar{K}}^{l+1}) \\ &< \tilde{Q}_i^{l+1}(\bar{\varepsilon}_{i(\infty)}^{(u_{i\infty}^{l+1}, u_{-i\infty}^{l+1})}, u_{i\infty}^{l+1}) - \tilde{Q}_i^{l+1}(\bar{\varepsilon}_{i\bar{K}}^{(u_{i\bar{K}}^{l+1}, u_{-i\bar{K}}^{l+1})}, u_{i\bar{K}}^{l+1}).\end{aligned}$$

Part a) implies that $\tilde{Q}_i^l(\bar{\varepsilon}_{i(\infty)}^{(u_{i\infty}^{l+1}, u_{-i\infty}^{l+1})}, u_{i\infty}^{l+1}) \to 0$ and $\tilde{Q}_i^{l+1}(\bar{\varepsilon}_{i(\infty)}^{(u_{i\infty}^{l+1}, u_{-i\infty}^{l+1})}, u_{i\infty}^{l+1}) \to 0$ such that

$$\tilde{Q}_i^{l+1}(\bar{\varepsilon}_{i\bar{K}}^{(u_{i\bar{K}}^{l+1}, u_{-i\bar{K}}^{l+1})}, u_{i\bar{K}}^{l+1}) < \tilde{Q}_i^l(\bar{\varepsilon}_{i\bar{K}}^{(u_{i\bar{K}}^{l+1}, u_{-i\bar{K}}^{l+1})}, u_{i\bar{K}}^{l+1}). \tag{77}$$

Therefore, by induction (77) yields

$$0 < \ldots < \tilde{Q}_i^{l+1} < \tilde{Q}_i^l < \ldots < \tilde{Q}_i^0, \ \forall i, l. \tag{78}$$

The stabilizing policies (65) form a Nash equilibrium tuple for the dynamic graphical game. The decreasing sequence (78) is bounded by $\{0, \tilde{Q}_i^0(\bar{\varepsilon}_{i0}^{(u_{i0}^0, u_0^0)}, u_{i0}^0)\}$. Then

the best response value function solutions $\tilde{Q}_i^*, \forall i$ exist and satisfy (39) according to Theorem 2 such that

$$0 < \ldots < \tilde{Q}_i^* \ldots < \tilde{Q}_i^{l+1} < \tilde{Q}_i^l < \ldots < \tilde{Q}_i^0, \ \forall i, l. \quad (79)$$

□

This result shows that Algorithm 1 converges when the performance indices are suitably chosen.

6 Critic network solutions for the graphical games

It is not clear how to best implement Algorithm 1. In the single-agent case the implementation details do not matter too much. Proper implementation of Algorithm 1 is needed for multi-agent graphical games where numerous agents are learning. There are different methods that are used to implement Policy Iteration Algorithm 1, these involve least squares, batch least squares, Actor-Critic neural network, etc. Herein, we present a novel method for implementing Algorithm 1 for multi-agent learning on graphs, wherein all agents learn simultaneously and computations are reduced. Algorithm 2 will be formulated such that it uses only critic networks. This is the easiest way to implement Algorithm 1, without the difficulties that are associated with the other methods. Algorithm 2 will use gradient descent to tune the weights of the critic at each iteration.

This section develops an online model-free critic network structure based on the Q-function value approximation (61) that is used to solve the dynamic graphical games in real-time. This is motivated by the graph games Algorithm 1. Each agent i has its own critic to perform the value evaluation using local information. The policy of each agent i is improved using (65).

The Q-function for each agent i $Q_i(\bar{\varepsilon}_{ik}, u_{ik})$ (61) is approximated by the critic neural network structure $\hat{Q}_i(\cdot|\tilde{W}_{ic})$, such that

$$\hat{Q}_{ik}(\cdot|\tilde{W}_{ic}) = \frac{1}{2} L_{ik}^{\mathrm{T}} \tilde{W}_{ic}^{\mathrm{T}} L_{ik}, \quad (80)$$

where $\tilde{W}_{ic} \in \mathbb{R}^{(nN_{i,j}+m_i)\times(nN_{i,j}+m_i)}, \forall i$ are the weights of the approximated structures $\hat{Q}_{ik}(\cdot|\tilde{W}_{ic}), \forall i$. $L_{ik} = [\varepsilon_{ik}^{\mathrm{T}} \ \cdots \ \varepsilon_{-ik}^{\mathrm{T}} \ \hat{u}_{ik}^{\mathrm{T}}]^{\mathrm{T}}$ is a vector of the state ε_{ik} and the control action approximation of each agent i $\hat{u}_{ik}$, and the states of its neighbors ε_{-ik}.

The control policy is given in terms of the structure $\hat{Q}_{ik}(\cdot|\tilde{W}_{ic})$ weights. The update of the control policy $\hat{u}_{ik}$ is given by (65) with $\tilde{H}_i = \tilde{W}_{ic}$ such that

$$\hat{u}_{ik} = -\tilde{W}_{ic(\hat{u}_{ik}\hat{u}_{ik})}^{-1} \tilde{W}_{ic(\hat{u}_{ik}\varepsilon_{ik})} \varepsilon_{ik} - \sum_{j\in N_i} \tilde{W}_{ic(\hat{u}_{ik}\hat{u}_{ik})}^{-1} \tilde{W}_{ic(\hat{u}_{ik}\varepsilon_{jk})} \varepsilon_{jk}, \quad (81)$$

where $\tilde{W}_{ic(\hat{u}_{ik}\varepsilon_{ik})}$ represents the block matrix defined by the positions of control action approximation and the state i.

The policy iteration Algorithm 1 requires that the approximation of the Q-function (80) to be written as

$$\hat{Q}_{ik}(\cdot|\bar{W}_{ic}) = \bar{W}_{ic}^{\mathrm{T}} \bar{L}_{ik}, \quad (82)$$

where $\bar{L}_i \in \mathbb{R}^{(nN_{i,j}+m_i)(nN_{i,j}+m_i+1)/2\times 1}$ denotes the Kronecker product quadratic polynomial basis vector with elements $\{(\bar{L}_i)_q(\bar{L}_i)_r\}_{q=1:(nN_{i,j}+m_i);\, r=1:(nN_{i,j}+m_i)}$ and $\bar{W}_{ic} = v(\tilde{W}_{ic})$ with $v(\cdot)$ a vector valued matrix function that acts on the symmetric matrices and returns a column vector.

Using (82), the Bellman equation (64) is written such that

$$\begin{aligned} &\bar{W}_{ic}^{\mathrm{T}}(\bar{L}_{ik} - \bar{L}_{i(k+1)}) \\ &= \frac{1}{2}(\varepsilon_{ik}^{\mathrm{T}} Q_{ii} \varepsilon_{ik} + \hat{u}_{ik}^{\mathrm{T}} R_{ii} \hat{u}_{ik} + \sum_{j\in N_i} \hat{u}_{jk}^{\mathrm{T}} R_{ij} \hat{u}_{jk}). \end{aligned} \quad (83)$$

The vectors $\bar{W}_{ic}, \forall i$ are calculated in real-time as will be shown below.

The critic network structure for each agent i performs the evaluation (64). The policy improvement (65) depends on the evaluated value function (64).

Let $\mathfrak{I}_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})}$ be the target value of the neural network structure $\tilde{Q}_i(\bar{\varepsilon}_{ik}, \hat{u}_{ik}) = (\hat{Q}_{ik}(\cdot|\tilde{W}_{ic}) - \hat{Q}_{i(k+1)}(\cdot|\tilde{W}_{ic}))$ such that

$$\mathfrak{I}_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})} = \frac{1}{2}(\varepsilon_{ik}^{\mathrm{T}} Q_{ii} \varepsilon_{ik} + \hat{u}_{ik}^{\mathrm{T}} R_{ii} \hat{u}_{ik} + \sum_{j\in N_i} \hat{u}_{jk}^{\mathrm{T}} R_{ij} \hat{u}_{jk}). \quad (84)$$

The neural network approximation error is given by

$$\xi_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})} = \mathfrak{I}_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},u_{ik})} - \tilde{Q}_i(\bar{\varepsilon}_{ik}, \hat{u}_{ik}). \quad (85)$$

The square sum of the approximation error for each neural network i can be written as

$$\begin{aligned} \mathrm{err}_i &= \frac{1}{2}(\xi_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})})^{\mathrm{T}} \xi_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},u_{ik})} \\ &= \frac{1}{2}\|\mathfrak{I}\mathfrak{I}_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})} - \bar{W}_{ic}^{\mathrm{T}} \mathfrak{I}(\bar{L}_{i(k,k+1)})\|_2^2, \end{aligned} \quad (86)$$

where $\mathfrak{I}\mathfrak{I}_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})} \in \mathbb{R}^{1\times(nN_{i,j}+m_i)(nN_{i,j}+m_i+1)/2}$ is a row vector of the target values (84) for $(nN_{i,j} + m_i)(nN_{i,j} + m_i + 1)/2$ samples, and $\mathfrak{I}(\bar{L}_{i(k,k+1)})$ is a square matrix of $(nN_{i,j} + m_i)(nN_{i,j} + m_i + 1)/2$ samples of $(\bar{L}_{ik} - \bar{L}_{i(k+1)})$.

The change in the critic neural network weights is given by gradient descent on this function whose gradient is

$$\begin{aligned}-\Delta\bar{W}_{ic}^{l\mathrm{T}} =&\left(\frac{\partial \mathrm{err}_i}{\partial \xi_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})}}\right)\left(\frac{\partial \xi_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})}}{\partial \bar{W}_{ic}^{\mathrm{T}}}\right)\Big|_{\bar{W}_{ic}^{\mathrm{T}}=\bar{W}_{ic}^{l\mathrm{T}}}\\ =&\left(\Im\Im_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})}-\bar{W}_{ic}^{l\mathrm{T}}\Im(\bar{L}_{i(k,k+1)})\right)\times\Im(\bar{L}_{i(k,k+1)})^{\mathrm{T}}.\end{aligned}\tag{87}$$

Therefore, the update rules for the network weights are given by

$$\begin{aligned}\bar{W}_{ic}^{(l+1)\mathrm{T}} =&\ \bar{W}_{ic}^{l\mathrm{T}}-\tilde{\mu}_{ic}\left(\Im\Im_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})}\right.\\ &\left.-\bar{W}_{ic}^{l\mathrm{T}}\Im(\bar{L}_{i(k,k+1)})\right)\Im(\bar{L}_{i(k,k+1)})^{\mathrm{T}},\end{aligned}\tag{88}$$

where $0<\tilde{\mu}_{ic}<1$ is the network-learning rate.

Remark 6 To solve for the weights $\bar{W}_{ic}, \forall i$ in the approximated Bellman equation (83), numerous instances of (83) must be obtained at successive time instants. For these equations to be independent, a persistence of excitation condition is needed. Our approach uses the gradient descent algorithm (87) to solve these equations, and hence a PE condition is also needed. This can be achieved by adding probing noise to the control, which is decayed to zero with time as the solution to (83) is learned.

6.1 Network weights online tuning in real-time

The following algorithm is used to tune the critic network weights in real-time using data measured along the system trajectories.

Algorithm 2 (Network weights online tuning)

1) Initialize the critic weights $\bar{W}_{ic}^{0}$.

2) Do Loop (l iterations).

2.1) Start with given initial states $\bar{\varepsilon}_{i0}\ \forall i$ on the system trajectory.

Do Loop (s iterations).

a) The network weights are given by $\bar{W}_{ic}^{s}=\bar{W}_{ic}^{l}, \forall i$.

b) Calculate $\hat{u}_i^s,\ \forall i$ using (81).

c) Measure the dynamics $\bar{\varepsilon}_{i(k+1)}^{s},\ \forall i$.

d) Evaluate the values $\tilde{Q}_i^s(\bar{\varepsilon}_{ik},\hat{u}_{ik}),\ \forall i$ using (82).

End Loop when $s=(nN_{i,j}+m_i)(nN_{i,j}+m_i+1)/2$.

2.2) Critic network weights update rule

$$\begin{aligned}\bar{W}_{ic}^{(l+1)\mathrm{T}} =&\bar{W}_{ic}^{l\mathrm{T}}-\tilde{\mu}_{ic}\left(\Im\Im_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})}\right.\\ &\left.-\bar{W}_{ic}^{l\mathrm{T}}\Im(\bar{L}_{i(k,k+1)})\right)\Im(\bar{L}_{i(k,k+1)})^{\mathrm{T}},\end{aligned}$$

where $\Im_{\bar{\varepsilon}_{ik}}^{\tilde{Q}_i(\bar{\varepsilon}_{ik},\hat{u}_{ik})}$ is given by (84).

2.3) On convergence of $\|\tilde{Q}_i^{l+1}-\tilde{Q}_i^{l}\|$ End, otherwise repeat steps 2.2) (l increment) and 2.3).

Remark 7 Algorithm 2 uses gradient descent technique to tune the critic network weights at each iteration. Theorem 3 proved the convergence of Algorithm 1 at each step. Assuming that the gradient descent algorithms converge exactly at each iteration, then Algorithm 2 at each step solves first the Bellman equation (64) and then the action update (65). Unfortunately, gradient descent cannot always be guaranteed to converge to the exact solutions in the approximation structures. However, simulations have shown the effectiveness of this algorithm.

6.2 Graphical game example and simulation results

In this section the graphical game problem is solved online in real-time using Algorithm 2. Simulations are performed to verify the proper performance of Algorithm 2.

Consider the directed graph with four agents shown in Fig. 1.

The data of the graph example are given as follows:

Agents' dynamics:

$$\mathrm{A}=\begin{bmatrix}0.995 & 0.09983\\ -0.09983 & 0.995\end{bmatrix},\ B_1=\begin{bmatrix}0.2047\\ 0.08984\end{bmatrix},$$

$$B_2=\begin{bmatrix}0.2147\\ 0.2895\end{bmatrix},\ B_3=\begin{bmatrix}0.2097\\ 0.1897\end{bmatrix},\ B_4=\begin{bmatrix}0.2\\ 0.1\end{bmatrix}.$$

Pinning gains: $g_1=0, g_2=0, g_3=0, g_4=1$.

Graph connectivity matrix: $e_{12}=0.8, e_{14}=0.7, e_{23}=0.6, e_{31}=0.8$.

Performance index weighting matrices: $Q_{11}=Q_{22}=Q_{33}=Q_{44}=I_{2\times 2}$, $R_{11}=R_{22}=R_{33}=R_{44}=1$, $R_{13}=R_{21}=R_{32}=R_{41}=0$, $R_{12}=R_{14}=R_{23}=R_{31}=1$.

The learning rates are $\tilde{\mu}_{ic}=0.0001, \forall i$.

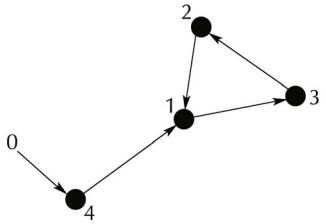

Fig. 1 Graphical game example.

Fig. 2 shows that, the neighborhood tracing error dynamics go to zero. Fig. 3 shows that, the dynamics of the agents synchronize to the leader while preserving the optimal behavior. Fig. 4 shows a 3D phase plane plot of the agents' dynamics. This figure shows that,

the agents synchronize to the leader agent's dynamics. These figures show that Algorithm 2 yields stability and synchronization to the leader's state. As shown in Remark 7, the gradient descent technique is assumed to converge at each step. Thus, a slow learning rate is chosen. The Policy Iteration Algorithm 2, guarantees the convergence of the agents to the leader's dynamics. For this graphical example, outer loop iterations in Algorithm 2 $l = 60$ to $l = 70$ are enough to maintain the synchronization.

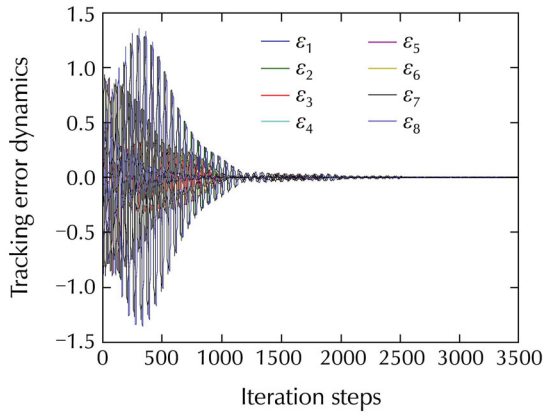

Fig. 2 Tracking error dynamics.

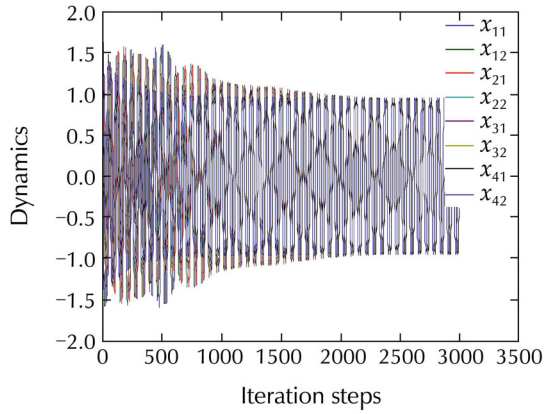

Fig. 3 Agents' dynamics.

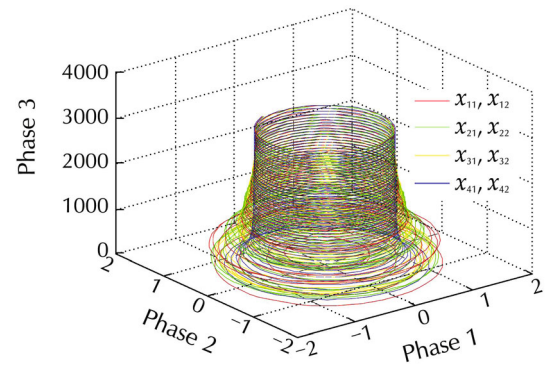

Fig. 4 Phase plot of the agents.

7 Conclusions

This paper studies a class of discrete-time dynamical games known as dynamic graphical games. Novel coupled Bellman equations and Hamiltonian functions are developed to solve the graphical game. Optimal control solutions for the dynamic graphical game are given in terms of the solutions to a set of coupled DTHJB equations. The stability and Nash solutions for the dynamic graphical game are proved. An online model-free policy iteration algorithm is developed to solve the dynamic graphical game in real-time. The developed algorithm does not require the knowledge of any of the agents' dynamics. Policy iteration convergence proof for the dynamic graphical game is given. A gradient descent technique with critic network structures is used to implement the online policy iteration algorithm to solve the dynamic graphical game.

References

[1] P. J. Werbos. Neural networks for control and system identification. *Proceedings of the 28th IEEE Conference on Decision and Control*, New York: IEEE, 1989: 260 – 265.

[2] P. J. Werbos. *Approximate Dynamic Programming for Real-time Control and Neural Modeling*. Handbook of Intelligent Control. D. A. White, D. A. Sofge (ed.). New York: Van Nostrand Reinhold, 1992.

[3] M. I. Abouheaf, F. L. Lewis, S. Haesaert, et al. Multi-agent discrete-time graphical games: interactive Nash equilibrium and value iteration solution. *Proceedings of the American Control Conference*, New York: IEEE, 2013: 4189 – 4195.

[4] K. G. Vamvoudakis, F. L. Lewis, G. R. Hudas. Multi-agent differential graphical games: online adaptive learning solution for synchronization with optimality. *Automatica*, 2012, 48(8): 1598 – 1611.

[5] A. E. Bryson. Optimal control-1950 to 1985. *IEEE Control Systems*, 1996, 16(3): 26 – 33.

[6] F. L. Lewis, D. Vrabie, V. L. Syrmos. *Optimal Control*. 3rd ed. New York: John Wiley & Sons, 2012.

[7] J. E. Marsden, M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 2001, 10(5): 357 – 514.

[8] Y. B. Suris. Discrete Lagrangian models. *International School on Discrete Integrable Systems*, Berlin: Springer, 2004: 111 – 184.

[9] S. Lall, M. West. Discrete variational Hamiltonian mechanics. *Journal of Physics A: Mathematical and General*, 2006, 39(19): 5509 – 5519.

[10] S. Mu, T. Chu, L. Wang. Coordinated collective motion in a motile particle group with a leader. *Physica A*, 2005, 351(2/4): 211 – 226.

[11] R. W. Beard, V. Stepanyan. Synchronization of information in distributed multiple vehicle coordinated control. *Proceedings of the IEEE Conference on Decision and Control*, Maui: IEEE, 2003: 2029 – 2034.

[12] A. Jadbabaie, J. Lin, A. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 2003, 48(6): 988 – 1001.

[13] R. Olfati-Saber, R. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 2004, 49(9): 1520 – 1533.

[14] Z. Qu. *Cooperative Control of Dynamical Systems: Applications to Autonomous Vehicles*. New York: Springer, 2009.

[15] W. Ren, R. Beard, E. Atkins. A survey of consensus problems in multi-agent coordination. *Proceedings of the American Control Conference*, New York: IEEE, 2005: 1859 – 1864.

[16] J. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. Ph.D. dissertation. Cambridge, MA: Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1984.

[17] Z. Li, Z. Duan, G. Chen, et al. Consensus of multi-agent systems and synchronization of complex networks: A unified viewpoint. *IEEE Transactions on Circuits and Systems*, 2010, 57(1): 213 – 224.

[18] X. Li, X. Wang, G. Chen. Pinning a complex dynamical network to its equilibrium. *IEEE Transactions on Circuits and Systems*, 2004, 51(10): 2074 – 2087.

[19] W. Ren, K. Moore, Y. Chen. High-order and model reference consensus algorithms in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement and Control*, 2007, 129(5): 678 – 688.

[20] J. Kuang, J. Zhu. On consensus protocols for high-order multiagent systems. *Journal of Control Theory and Applications*, 2010, 8(4): 406 – 412.

[21] S. Zhang, G. Duan. Consensus seeking in multi-agent cooperative control systems with bounded control input. *Journal of Control Theory and Applications*, 2011, 9(2): 210 – 214.

[22] R. Gopalakrishnan, J. R. Marden, A. Wierman. An architectural view of game theoretic control. *Performance Evaluation Review*, 2011, 38(3): 31 – 36.

[23] T. Başar, G. J. Olsder. *Dynamic Non-cooperative Game Theory*. Classics in Applied Mathematics. 2nd ed. Philadelphia: SIAM, 1999.

[24] G. Freiling, G. Jank, H. Abou-Kandil. On global existence of Solutions to coupled matrix Riccati equations in closed-loop Nash Games. *IEEE Transactions on Automatic Control*, 2002, 41(2): 264 – 269.

[25] Z. Gajic, T.-Y. Li. Simulation results for two new algorithms for solving coupled algebraic Riccati equations. *Proceedings of the 3rd International Symposium on Differential Games*. Sophia Antipolis, France, 1988.

[26] A. G. Barto, R. S. Sutton, C. W. Anderson. Neuron like adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems Man and Cybernetics*, 1983, 13(5): 834 – 846.

[27] R. Howard. *Dynamic Programming and Markov Processes*. Cambridge, MA: MIT Press, 1960.

[28] R. Bellman. *Dynamic Programming*. Princeton: Princeton University Press, 1957.

[29] D. P. Bertsekas, J. N. Tsitsiklis. *Neuro-dynamic Programming*. Belmont, MA: Athena Scientific 1996.

[30] P. J. Werbos. Intelligence in the Brain: a theory of how it works and how to build it. *Conference on Goal-Directed Neural Systems*, Oxford: Pergamon-Elsevier Science Ltd., 2009: 200 – 212.

[31] D. Vrabie, F. L. Lewis. Adaptive dynamic programming for online solution of a zero-sum differential game. *Journal of Control Theory and Applications*, 2011, 9(3): 353 – 360.

[32] J. Morimoto, G. Zeglin, C. Atkeson. Minimax differential dynamic programming: application to a biped walking robot. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, New York: IEEE, 2003: 1927 – 1932.

[33] T. Landelius. *Reinforcement Learning and Distributed Local Model Synthesis*. Ph.D. dissertation. Sweden: Linkoping University, 1997.

[34] R. S. Sutton, A. G. Barto. *Reinforcement Learning - An Introduction*. Cambridge, MA: MIT Press, 1998.

[35] S. Sen, G. Weiss. *Learning In Multiagent Systems, in Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA: MIT Press, 1999: 259 – 298.

[36] K. G. Vamvoudakis, F. L. Lewis. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 2010, 46(5): 878 – 888.

[37] K. G. Vamvoudakis, F. L. Lewis. Multi-player non-zero sum games: online adaptive learning solution of coupled Hamilton-Jacobi equations. *Automatica*, 2011, 47(8): 1556 – 1569.

[38] D. Vrabie, O. Pastravanu, F. L. Lewis, et al. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica*, 2009, 45(2): 477 – 484.

[39] D. P. Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 2011, 9(3): 310 – 335.

[40] L. Busoniu, R. Babuska, B. De-Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man and Cybernetics*, 2008, 38(2): 156 – 172.

[41] P. Vrancx, K. Verbeeck, A. Nowe. Decentralized learning in markov games. *IEEE Transactions on Systems, Man and Cybernetics*, 2008, 38(4): 976 – 981.

[42] M. L. Littman. Value-function reinforcement learning in Markov games. *Cognitive Systems Research*, 2001, 2(1): 55 – 66.

[43] Y. Jiang, Z. Jiang. Computational adaptive optimal control for continuous-time linear systems with completely unknown dynamics. *Automatica*, 2012, 48(10): 2699 – 2704.

[44] Y. Jiang, Z. Jiang. Global adaptive dynamic programming for continuous-time nonlinear systems. 2013: http://arxiv.org/abs/1401.0020.

[45] T. Dierks, S. Jagannathan. Optimal control of affine nonlinear continuous-time systems using an online Hamilton-Jacobi-Isaacs formulation. *Proceedings of the 49th IEEE Conference on Decision and Control*, New York: IEEE, 2010: 3048 – 3053.

[46] M. Johnson, T. Hiramatsu, N. Fitz-Coy, et al. Asymptotic Stackelberg optimal control design for an uncertain Euler Lagrange system. *Proceedings of the 49th IEEE Conference on Decision and Control*, New York: IEEE, 2010: 6686 – 6691.

[47] F. L. Lewis. *Applied Optimal Control and Estimation: Digital Design and Implementation*. Englewood Cliffs: Prentice Hall, 1992.

[48] S. Khoo, L. Xie, Z. Man. Robust finite-time consensus tracking algorithm for multirobot systems. *IEEE/ASME Transactions on Mechatronics*, 2009, 14(2): 219 – 228.

Mohammed I. ABOUHEAF was born in Smanoud, Egypt. He received his B.Sc. and M.Sc. degrees in Electronics and Communication Engineering, Mansoura College of Engineering, Mansoura, Egypt, in 2000 and 2006, respectively. He worked as an assistant lecturer with the Air Defense College, Alexandria, Egypt (2001-2002). He worked as a planning engineer for the Maintenance Department, Suez Oil Company (SUCO), South Sinai, Egypt (2002-2004). He worked as an assistant lecturer with the Electrical Engineering Department, Aswan College of Energy Engineering, Aswan, Egypt (2004-2008). He received his Ph.D. degree in Electrical Engineering, University of Texas at Arlington (UTA), Arlington, Texas, U.S.A. in 2012. He worked as a postdoctoral fellow with the University of Texas at Arlington Research Institute (UTARI), Fort Worth, Texas, U.S.A. (2012-2013). He worked as Adjunct Faculty with the Electrical Engineering Department, University of Texas at Arlington (UTA), Arlington, Texas, U.S.A. (2012-2013). He was a member of the Advanced Controls and Sensor Group (ACS) and the Energy Systems Research Center (ESRC), University of Texas at Arlington, Arlington, Texas, U.S.A. (2008-2012). Currently, he is Assistant Professor with the Systems Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. His research interests include optimal control, adaptive control, reinforcement learning, fuzzy systems, game theory, microgrids, and economic dispatch. E-mail: abouheaf@kfupm.edu.sa.

Frank L. LEWIS is Member, National Academy of Inventors. Fellow IEEE, Fellow IFAC, Fellow, U.K. Institute of Measurement & Control, PE Texas, U.K. Chartered Engineer, UTA Distinguished Scholar Professor, UTA Distinguished Teaching Professor, and Moncrief-O'Donnell Chair at the University of Texas at Arlington Research Institute. He is Qian Ren Thousand Talents Consulting Professor, Northeastern University, Shenyang, China. He obtained the B.S. degree in Physics/EE and the MSEE at Rice University, the M.S. in Aeronautical Engineering from University of West Florida, and the Ph.D. degree at Georgia Institute of Technology. He works in feedback control, intelligent systems, cooperative control systems, and nonlinear systems. He is Author of 6 U.S. patents, numerous journal special issues, journal papers, and 20 books, including Optimal Control, Aircraft Control, Optimal Estimation, and Robot Manipulator Control which are used as university textbooks worldwide. He received the Fulbright Research Award, NSF Research Initiation Grant, ASEE Terman Award, Int. Neural Network Soc. Gabor Award, U.K. Inst Measurement & Control Honeywell Field Engineering Medal, and IEEE Computational Intelligence Society Neural Networks Pioneer Award. He received Outstanding Service Award from Dallas IEEE Section, and selected as Engineer of the year by Ft. Worth IEEE Section. He was listed in Ft. Worth Business Press Top 200 Leaders in Manufacturing. He received Texas Regents Outstanding Teaching Award 2013. He is Distinguished Visiting Professor at Nanjing University of Science & Technology and Project 111 Professor at Northeastern University in Shenyang, China. He is Founding Member of the Board of Governors of the Mediterranean Control Association. E-mail: lewis@uta.edu.

Magdi S. MAHMOUD obtained the B.Sc. degree (Honors) in Communication Engineering, M.Sc. degree in Electronic Engineering, and Ph.D. degree in Systems Engineering, all from Cairo University in 1968, 1972 and 1974, respectively. He has been a professor of Engineering since 1984. He is now a distinguished professor at KFUPM, Saudi Arabia. He was on the faculty at different universities worldwide including Egypt (CU, AUC), Kuwait (KU), U.A.E. (UAEU), U.K. (UMIST), U.S.A. (Pitt, Case Western), Singapore (Nanyang) and Australia (Adelaide). He lectured in Venezuela (Caracas), Germany (Hanover), U.K. ((Kent), U.S.A. (UoSA), Canada (Montreal) and China (BIT, Yanshan). He is the principal author of thirty-four (34) books, inclusive book-chapters and the author/co-author of more than 510 peer-reviewed papers. He is the recipient of two national, one regional and four university prizes for outstanding research in engineering and applied mathematics. He is a fellow of the IEE, a senior member of the IEEE, the CEI (U.K.), and a registered consultant engineer of information engineering and systems (Egypt). He is currently actively engaged in teaching and research in the development of modern methodologies to distributed control and filtering, networked-control systems, triggering mechanisms in dynamical systems, fault-tolerant systems and information technology. He is a fellow of the IEEE, a senior member of the IEEE, the CEI (U.K.), and a registered consultant engineer of information engineering and systems Egypt. E-mail: msmahmoud@kfupm.edu.sa, magdisadekmahmoud@gmail.com, magdim@yahoo.com.

Dariusz MIKULSKI is a research computer scientist in Ground Vehicle Robotics at the U.S. Army Tank-Automotive Research Development and Engineering Center in Warren, MI. He currently works on research to improve cooperative teaming and cyber security in military unmanned convoy operations. Dr. Mikulski earned his Ph.D. degree in Electrical and Computer Engineering at Oakland University in Rochester Hills, Michigan in 2013. He also earned his B.Sc. in Computer Science from the University of Michigan in Ann Arbor and Masters in Computer Science and Engineering from Oakland University. E-mail: dariusz.g.mikulski.civ@mail.mil.